Static knowledge-compilation lower bounds for Maroti-pair CSPs

Filip Stappers

May 26, 2026

Abstract

We ask whether CSPs preserved by a finite idempotent Maroti pair admit a single polynomial-size static representation of their full solution set. These CSPs are already polynomial-time decidable: every finite idempotent Maroti pair is Taylor, so Bulatov–Zhuk applies [5, 20]. The question is whether this tractability also comes with a general compact representation theorem.

Our main result is negative for the standard knowledge-compilation languages. We exhibit four explicit computer-verified Maroti algebras and show that, for each of them, some family of arity-at-most-two CSP instances has solution sets of exponential size in every fixed language $\mathcal{L} \subseteq$ DNNF, including OBDD, FBDD, SDD, d-DNNF, and DNNF itself. The proof embeds Bova–Capelli–Mengel–Slivovsky expander CNFs [3] into the CSPs of these algebras through verified invariant binary relations on two-element subuniverses. Thus no fixed standard KC language gives a universal polynomial-size representation method for all Maroti-pair CSPs. Section 7 upgrades the witness-based result to a universal lower bound covering every finite idempotent Maroti pair without a cube term, by replacing the two-element subuniverse hypothesis with the strictly weaker hypothesis "has a strict cube-term blocker on some subuniverse," which always holds in the residual blocker regime.

Beyond standard KC, we extend the negative answer to two broader exact-static circuit classes by a direct fully-induced-matching counting argument on the BCMS expander cuts: every exposed-interface exact-summary circuit (in tree or DAG form) and every strict-parametric symbolic template DAG representing the same witness families also has size $2^{\Omega(n)}$. The remaining symbolic-circuit escape requires either growing parameter arity, context-dependent resolution, or super-polynomial symbolic alphabets.

On the positive side, bounded-treewidth instances still admit the standard exact dynamic-programming compilation of size $|A|^{O(w)}\mathrm{poly}(|\Theta|)$.

1 Introduction

Many tractable CSP classes admit reusable static representations of their full solution sets. Bounded-width templates admit projection-based compactness, Mal'tsev templates admit compact generating-set representations, and the few-subpowers setting gives the classical k-edge representation theorems. In all of these cases, polynomial-time solvability comes with some general-purpose static structure theorem.

This paper asks the corresponding question for finite idempotent Maroti pairs (defined formally in Section 2). These algebras are polynomial-time tractable by the Bulatov–Zhuk dichotomy [5, 20], but they sit outside the standard named compact representation regimes. So the issue is not decision complexity. The issue is whether one can compile every such CSP instance into a reusable polynomial-size object representing its exact solution set.

The initial motivation for our investigation was exercise 474 of Knuth posed in [14]:

> Is there a procedure analogous to Algorithms M and N that exploits two polymorphisms $p(x, y, z)$ and $q(x, y, z)$ when p and q form a Maroti pair?

Here Algorithm M refers to the bounded-width / near-unanimity projection method, and Algorithm N to the Mal'tsev compact-representation method. In the language of this paper, the question is whether there exists a single polynomial-size static representation method for solution sets of CSP instances preserved by an arbitrary finite idempotent Maroti pair.

Our answer is negative in the standard knowledge-compilation setting. We prove this using four explicit witness algebras. For every fixed standard knowledge- compilation language $\mathcal{L} \subseteq \mathrm{DNNF}$, there is no universal polynomial-size $\mathcal{L}$-compilation scheme for Maroti-pair CSPs. The witness argument extends to the entire residual blocker regime via the R_{abs} generalization in Section 7. Section 9 extends the negative answer further: every exposed-interface exact-summary circuit (in tree or DAG form) and every strict-parametric symbolic template DAG representing the same witness families also has size $2^{\Omega(n)}$, by a direct fully-induced-matching counting argument on balanced expander cuts. Bounded-treewidth instances still admit the usual exact dynamic-programming compilation, and beyond all of the above the only exact-static frontier still left in view is genuinely context-sensitive symbolic compression.

2 Preliminaries

Definition 1 (Maroti pair)**.** A finite idempotent algebra $A = (A; p, q)$ is a *Maroti pair* if p and q are ternary idempotent operations satisfying

$$\begin{aligned} &\text{(M1)} \quad p(x, y, x) \approx q(x, y, x), \\ &\text{(M2)} \quad p(x, y, y) \approx q(x, x, y), \\ &\text{(M3)} \quad q(x, x, y) \approx q(y, x, x). \end{aligned}$$

These identities do not imply majority or near-unanimity behavior. In particular, setting $p = q = m$ with $m(x, y, z) = x - y + z$ does *not* produce a Maroti pair, since (M2) would collapse to $x \approx y$.

For a finite algebra A, let $\mathrm{CSP}(A)$ denote the class of CSP instances whose constraint relations are invariant under the term clone $\mathrm{Clo}(A)$ of all finitary term operations of A; equivalently, whose constraint relations lie in the relational clone $\mathrm{Inv}(A)$.

We use *Taylor algebra*, *near-unanimity term*, *Mal'tsev term*, *semilattice term*, and *k-edge term* in their standard universal-algebraic senses. In particular, a Taylor algebra is one admitting a Taylor term, and a k-edge term is the standard Idziak–Marković–McKenzie–Valeriote–Willard few-subpowers term condition; no explicit identities for these notions are needed below.[1]

Definition 2 (Static compilation scheme)**.** A *representation language* $\mathcal{L}$ assigns to each finite relation $R \subseteq D^n$ a finite object $C \in \mathcal{L}(D, n)$ intended to encode R, together with a membership test that runs in time polynomial in $|C| + n$ on input a tuple of D^n. An *$\mathcal{L}$-static compilation scheme* for a class of CSP instances is a polynomial-time compiler that maps each instance Θ to an $\mathcal{L}$-representation of its exact solution relation $\mathrm{Sol}(\Theta)$, of size polynomial in $|\Theta|$.

In this paper $\mathcal{L}$ is always a fixed standard knowledge-compilation language in the sense of Darwiche–Marquis [7]. Thus the negative result concerns uniform compilation into a fixed target language, not merely nonuniform polynomial-time solvability.

[1] For orientation: a Mal'tsev term satisfies $m(x, y, y) \approx m(y, y, x) \approx x$, a near-unanimity term satisfies $n(y, x, \ldots, x) \approx \cdots \approx n(x, \ldots, x, y) \approx x$, a semilattice term is an idempotent commutative associative binary term, and a k-edge term is the standard few-subpowers term condition from [12].

We identify a finite relation $R \subseteq D^n$ with its indicator function $\mathbf{1}_R : D^n \to \{0,1\}$. When comparing with Boolean KC lower bounds, we pass from $\mathbf{1}_R$ to its standard Boolean encoding over variables that record the chosen value in each coordinate.

3 Regimes for Maroti Pairs

The first structural ingredient is the familiar regime split inherited from the Taylor dichotomy and tame-congruence theory.

Proposition 3 (Relevant regime split)**.** *For the purposes of this paper, the relevant finite idempotent Taylor regimes for a finite idempotent Maroti pair are the following cases:*

(1) the bounded-width / SD(∧) regime,

(2) the cube-term regime, equivalently the few-subpowers regime in the finite idempotent setting,

(3) the residual regime outside bounded width and few subpowers, characterized by the presence of type 2 and absence of a cube term.

Proof sketch. The Maroti identities rule out all projection assignments to p and q, so a finite idempotent Maroti pair is Taylor. In the finite idempotent CSP setting, the Taylor side omits type 1, and omission of type 2 as well is exactly the SD(∧)/bounded-width side; see Hobby–McKenzie for tame-congruence-theoretic background and Barto–Kozik for the idempotent CSP formulation [11, 2]. If type 2 occurs and the algebra has a cube term, then the standard few-subpowers theorem applies [12]. If type 2 occurs and no cube term exists, the finite idempotent cube-term-blocker theorem places the algebra in the residual blocker regime. These are the three cases needed for the present discussion, although cases (1) and (2) may overlap in general [16]. □

The witness argument then works inside the following narrower simple subclass.

Definition 4 (Simple witness regime)**.** In the witness-based sections below, a finite simple Maroti pair is said to lie in the *simple witness regime* if it is in case (3) of Proposition 3 and has no global Mal'tsev term, no global near-unanimity term, no global semilattice term, and no global k-edge term. Section 7 returns to the full residual blocker regime, without this simplicity restriction.

4 Earlier Work

The proof draws on four standard ingredients from the algebraic-CSP and knowledge-compilation literatures. Here SD(∧) denotes the congruence meet-semidistributive class SD(∧).

(1) Taylor identities and tame-congruence theory give the standard regime split for finite idempotent templates: bounded width on the SD(∧) side, few-subpowers on the cube-term side, and the residual blocker regime otherwise [11, 2, 12].

(2) Projection-based compactness for near-unanimity algebras (Algorithm M), Mal'tsev representations (Algorithm N), and k-edge / few-subpowers representations supply the usual positive template families one would test first for Maroti pairs [1, 4, 12].

(3) The Bulatov and Zhuk dichotomy theorems establish polynomial-time solvability for CSPs over finite Taylor templates, including all finite idempotent Maroti pairs [5, 20].

(4) Bova–Capelli–Mengel–Slivovsky prove exponential DNNF lower bounds for expander-CNF families. Combined with verified invariant binary relations on specific two-element subuniverses of specific Maroti algebras, these lower bounds transfer directly to bounded-arity CSP solution sets [3].

We combine these ingredients into a witness-based transfer theorem showing that universal static compilation already fails on four concrete Maroti algebras.

In the simple witness regime, none of the named compactness theorems from ingredient (2) above applies: the near-unanimity projection-storage theorem requires a near-unanimity term, Mal'tsev compactness requires a Mal'tsev term, and few-subpowers compactness requires a k-edge term, all of which are excluded by definition of the regime. Simplicity also rules out any nontrivial semilattice-of-Mal'tsev-blocks decomposition (no nontrivial congruences). This only says that the standard compactness theorems do not apply; it does not rule out every other compactness mechanism. The stronger exclusion proved below concerns the standard KC languages.

5 The four witness algebras

Appendix A records four verified simple Maroti algebras, which we denote by

$$A_{\wedge,5}, \qquad A_{\vee,5}, \qquad A_{\wedge,7}, \qquad A_{\vee,7}.$$

For each of them, a two-element subuniverse $B = \{b_0, b_1\}$ and a binary relation R_{OR} or R_{NAND} preserved by $\mathrm{Clo}(A)$ were verified explicitly by truth-table computation. Appendix A records the full catalog, including the restricted operation tables on B.

The witness verifications run as follows. For $A_{\wedge,5}$ on $B = \{0, 4\}$, both restricted generators are Boolean AND, so R_{NAND} is invariant. For $A_{\vee,5}$ on $B = \{2, 3\}$ and $A_{\vee,7}$ on $B = \{4, 6\}$, the restricted operation $q|_B$ is Boolean OR, while $p|_B$ differs from Boolean OR at one input but still preserves R_{OR} by direct check. For $A_{\wedge,7}$ on $B = \{0, 1\}$, the restricted generators are not literally the AND clone. Invariance of R_{NAND} in this case follows from a *dominant-projection* check: under the identification $0 \leftrightarrow 0$, $1 \leftrightarrow 1$, each restricted generator $f|_B : B^3 \to B$ is pointwise bounded above by a fixed coordinate projection $\pi_{i_f}(x_1, x_2, x_3) = x_{i_f}$ (read off the eight-bit table: $p|_B = \texttt{00001001} \leq \pi_1$ and $q|_B = \texttt{00010011} \leq \pi_2$, both verified coordinatewise on the eight inputs). Since R_{NAND} is downward-closed in the product order on B^2 and each coordinate projection preserves R_{NAND} trivially, any operation pointwise bounded above by a coordinate projection also preserves R_{NAND}. Thus the witness-family form of the lower-bound transfer depends only on four small verified two-element restrictions. The universal form in Section 7 relaxes the two-element hypothesis to "has a strict cube-term blocker on some subuniverse," using the $R_{\mathrm{abs}}^{(D,S)}$ generalization of R_{NAND}.

The next sections develop the lower-bound chain. Section 6 proves the witness-family lower bound for standard KC languages; Section 7 upgrades it to a universal lower bound covering every finite idempotent Maroti pair without a cube term; Section 8 records the parameterized positive slice for bounded treewidth; Section 9 extends the negative answer beyond standard KC with unconditional $2^{\Omega(n)}$ lower bounds for exposed-interface exact-summary tree, exposed-interface exact-summary DAG, and strict-parametric template circuit classes.

6 A General Lower Bound for Maroti Pairs

We now state the main representation-theoretic lower bound. We first isolate the representation-side reduction as a stand-alone lemma, since it can also be used as a necessary criterion for any candidate representation.

6.1 KC-reducibility lemma

The main theorem combines two ingredients: an algebraic construction placing an expander-CNF instance inside a CSP over a witness algebra from the simple witness regime, and a reduction that turns any sufficiently structured exact representation of that CSP back into a Boolean DNNF of comparable size.

Lemma 5 (KC-reducibility)**.** *Let A be a finite algebra with a verified two-element subuniverse $B = \{b_0, b_1\} \subseteq A$ on which the restricted term clone preserves a Boolean-gadget relation R_{OR} or R_{NAND}, and let $R \subseteq B^n \subseteq A^n$ be the solution set of an arity-at-most-two CSP over A obtained from the gadget encoding. Let $\mathcal{R}$ be an exact-static AND/OR representation of R whose leaves are value tests $x_i = a$ for values $a \in A$. If $\mathcal{R}$ satisfies*

(D1) decomposability: *the representation has the standard Darwiche–Marquis decomposability property, namely that every conjunction step combines subrepresentations on disjoint variable sets;*

(D2) per-coordinate value restriction closure: *replacing the allowed value set of any coordinate by a subset of A yields an exact representation of the restricted relation of size at most $O(|\mathcal{R}|)$;*

(D3) polynomial-time membership: *tuple membership in the represented relation is decidable in time polynomial in $|\mathcal{R}| + n$;*

then there exists a Boolean DNNF *D computing the same relation viewed over B^n with $|D| = O(|\mathcal{R}|)$.*

Proposition 6 (Target languages satisfy (D1)–(D3))**.** *The languages* DNNF*, d-DNNF, SDD, FBDD, and OBDD are Boolean-literal AND/OR languages in the sense of [7]. The finite-domain A-valued analog of each, in which Boolean literals are replaced by value-test leaves $x_i = a$, satisfies* (D1)–(D3)*:* (D1) *holds because each such language is by definition an AND/OR circuit with leaves of the required form and decomposable conjunctions;* (D2) *is the standard conditioning operation on unary value tests, with cost at most constant in each language [7];* (D3) *is the standard polynomial-time model-checking guarantee. Boolean DNNF representations over the one-hot or bit encoding of A-valued variables are bit-predicate languages rather than value-test languages, but the substitution paragraph following Lemma 12 translates them into the value-test setting at constant overhead per leaf, so the lemma applies to those Boolean representations as well.*

Proof sketch. By (D2), restrict the domain of every coordinate to B at constant-factor cost. This removes every leaf testing a value outside B. Each surviving leaf is now a test $x_i = b_0$ or $x_i = b_1$, which we relabel as $\neg x_i$ or x_i, respectively. Since $R \subseteq B^n$ by hypothesis, this restriction leaves the represented relation equal to R; it only shrinks leaf domains. By (D1) the resulting object is a Boolean DNNF. Because $\mathcal{R}$ is exact and the relabeling preserves the allowed tuples on B, acceptance of D agrees pointwise with membership in R. Each step changes size by at most a constant factor, so $|D| = O(|\mathcal{R}|)$. □

Corollary 7 (KC-reducibility corollary)**.** *Let $\{\Theta_n\}$ be a family of arity-at-most-two CSP instances over a finite algebra A, obtained by encoding a Boolean family $\{F_n\}$ through a verified two-element subuniverse $B = \{b_0, b_1\}$ and an invariant relation R_{OR} or R_{NAND}, so that* $\mathrm{Sol}(\Theta_n) \subseteq B^n$ *is in bijection with* $\mathrm{Sat}(F_n)$*. If every Boolean* DNNF *for* $\mathrm{Sat}(F_n)$ *has size $2^{\Omega(n)}$, then every exact-static representation of* $\mathrm{Sol}(\Theta_n)$ *in a target language satisfying* (D1)–(D3) *has size $2^{\Omega(n)}$.*

Proof. Otherwise, for infinitely many n there would be a representation of $\mathrm{Sol}(\Theta_n)$ of subexponential size in such a target language. Lemma 5 together with Proposition 6 would convert it into a Boolean DNNF for $\mathrm{Sat}(F_n)$ of comparable size, contradicting the assumed lower bound. □

Theorem 8 (Witness-family lower bound)**.** *For each of the four witness algebras $A_{\wedge,5}$, $A_{\vee,5}$, $A_{\wedge,7}$, and $A_{\vee,7}$, there is a family $\{\Theta_n^A\}_{n\geq 1}$ of arity-at-most-two CSP instances over A of size polynomial in n such that every* DNNF *representation of the solution set has size $2^{\Omega(n)}$. Consequently the same bound holds for OBDD, FBDD, SDD, d-DNNF, and every standard knowledge-compilation language $\mathcal{L} \subseteq$ DNNF.*

Proof. We give the argument in detail for the witness algebra $A = A_{\wedge,5}$ with $B = \{0, 4\}$ and verified invariant relation $R_{\mathrm{NAND}} = \{(0,0),(0,4),(4,0)\} \subseteq B^2$; the other three witnesses (Appendix A) work identically with R_{OR} or R_{NAND} and the corresponding two-element subuniverse.

Let G_n be the bounded-degree expander on n vertices from the Bova–Capelli–Mengel–Slivovsky family [3], and let F_n be the monotone NAND-style 2-CNF $\bigwedge_{(u,v)\in E(G_n)} \neg(x_u \wedge x_v)$ on Boolean variables $x_1, \ldots, x_n$, whose satisfying assignments are exactly the indicator vectors of independent sets in G_n.

Construct the CSP Θ_n^A with n variables $v_1, \ldots, v_n$ each over $A = \{0, 1, 2, 3, 4\}$, and one binary constraint $(v_u, v_v) \in R_{\mathrm{NAND}}$ for each edge $(u, v) \in E(G_n)$. This is an arity-at-most-two CSP of size linear in $|F_n| + n$. Because $R_{\mathrm{NAND}} \subseteq B^2$ and G_n has no isolated vertices, every constraint forces both its variables to lie in B; hence $\mathrm{Sol}(\Theta_n^A) \subseteq B^n$. The identification $0 \leftrightarrow 0_{\mathrm{Bool}}$, $4 \leftrightarrow 1_{\mathrm{Bool}}$ sends $\mathrm{Sol}(\Theta_n^A)$ bijectively onto $\mathrm{Sat}(F_n)$: $(v_u, v_v) \in R_{\mathrm{NAND}}$ iff not both endpoints take value 4, which is exactly the NAND clause $\neg(x_u \wedge x_v)$.

Let $\mathcal{R}$ be any representation of $\mathrm{Sol}(\Theta_n^A)$ in a target language $\mathcal{L} \subseteq$ DNNF with value-test leaves. By Corollary 7 (whose hypotheses are checked for $\mathcal{L}$ by Proposition 6), it suffices to exhibit a Boolean DNNF for $\mathrm{Sat}(F_n)$ of size $O(|\mathcal{R}|)$. Apply property (D2) of Lemma 5 to restrict every coordinate from A to B at constant overhead per gate: every leaf testing $v_i = a$ with $a \in \{1, 2, 3\}$ becomes the constant FALSE, and dually $v_i \neq a$ becomes TRUE. The surviving leaves test $v_i = 0$ or $v_i = 4$; relabel them as Boolean literals $\neg x_i$ and x_i respectively (and dually for the negated forms). The result is a Boolean DNNF D on variables $x_1, \ldots, x_n$ whose accepted set, under the identification above, equals $\mathrm{Sol}(\Theta_n^A)$, which is exactly $\mathrm{Sat}(F_n)$. Each step changes size by at most a constant factor, so $|D| = O(|\mathcal{R}|)$.

The BCMS lower bound [3] forces $|D| \geq 2^{\Omega(n)}$, hence $|\mathcal{R}| \geq 2^{\Omega(n)}$, and the same bound holds in every standard knowledge-compilation language $\mathcal{L} \subseteq$ DNNF by Proposition 6.

For $A_{\wedge,7}$, the restricted operation tables are not literally the AND clone, but the dominant-projection check recorded in Appendix A verifies that R_{NAND} is still preserved on $B = \{0, 1\}$; the rest of the argument is unchanged with the identification $0 \leftrightarrow 0_{\mathrm{Bool}}$, $1 \leftrightarrow 1_{\mathrm{Bool}}$. For $A_{\vee,5}$ and $A_{\vee,7}$, replace R_{NAND} by $R_{\mathrm{OR}} = \{(b_0, b_1), (b_1, b_0), (b_1, b_1)\}$ and F_n by the monotone OR-style 2-CNF $\bigwedge_{(u,v)\in E(G_n)}(x_u \vee x_v)$, whose satisfying assignments are vertex covers of G_n (the bit-complement of the independent-set encoding); the BCMS lower bound applies in either form. □

Corollary 9 (No universal KC-family compiler)**.** *Fix any standard knowledge-compilation language $\mathcal{L}$ contained in* DNNF*. There is no universal $\mathcal{L}$-static compilation scheme for the class of all finite idempotent Maroti-pair CSP instances, even when restricted to arity-at-most-two instances.*

Proof. Any such compiler would in particular have to compile CSP instances over the four witness algebras, and these witnesses already lie inside the class of finite idempotent Maroti-pair CSP instances. The theorem above gives a family of instances for each of those algebras whose exact solution relation has exponential size in every DNNF representation, hence also in every language $\mathcal{L} \subseteq$ DNNF. This contradicts the assumed polynomial-size bound for an $\mathcal{L}$-static compilation scheme. The point is not decision complexity: the same class remains polynomial-time decidable by Bulatov–Zhuk. □

Remark 10 (Natural instances versus witness families). Corollary 7 makes precise what counts as representation-theoretic evidence for or against a Maroti compression candidate. Empirical performance on naturally occurring or small "real" Maroti instances provides only limited evidence: only experiments on $\{\Theta_n^A\}$ for the four witness algebras at growing n test whether a candidate evades the lower bound, since these are the instances on which the theorem applies. A candidate that claims compression on a different family of natural instances does not bear on the universal-static question unless it is also measured on Θ_n^A.

7 The Full Residual-Blocker Theorem

The witness theorem of Section 6 is logically sufficient for the negative answer. We now upgrade it to a universal statement covering *every* finite idempotent Maroti pair without a cube term, by replacing the two-element subuniverse hypothesis used in Lemma 5 with the strictly weaker hypothesis "has a strict cube-term blocker on some subuniverse."

7.1 Generalized invariant gadget

For a finite idempotent algebra A and a strict Marković–Maróti cube-term blocker (D, S) of A (so $D, S \leq A$ are subuniverses, $\emptyset \neq D \subsetneq S$, and every A-term has a D-absorbing position on S^n) define

$$R_{\text{abs}}^{(D,S)} := \{(x, y) \in S^2 : x \in D \text{ or } y \in D\}.$$

This is the natural generalization of R_{NAND} from a two-element subuniverse to an arbitrary blocker carrier.

Lemma 11 (Generalized Bridge endpoint)**.** *Let (D, S) be a strict cube-term blocker of A. Then $R_{\text{abs}}^{(D,S)}$ is $\text{Clo}(A)$-invariant.*

Proof. Let $g \in \text{Clo}(A)$ be n-ary. By the strict blocker hypothesis, g has a D-absorbing position $i_g \in [n]$ on S: whenever $\bar{s} \in S^n$ and $s_{i_g} \in D$, $g(\bar{s}) \in D$.

Take tuples $t_1, \ldots, t_n \in R_{\text{abs}}^{(D,S)}$ and let $r := (g(t_1[1], \ldots, t_n[1]), g(t_1[2], \ldots, t_n[2])) \in S^2$. Suppose $r \notin R_{\text{abs}}^{(D,S)}$, so both coordinates of r lie in $S \setminus D$. Then by the absorbing-position property applied coordinatewise, $t_{i_g}[1] \notin D$ and $t_{i_g}[2] \notin D$, hence t_{i_g} has no D-coordinate. But $t_{i_g} \in R_{\text{abs}}^{(D,S)}$, contradiction. □

7.2 Generalized KC-reducibility

For a monotone OR-CNF F on Boolean variables $x_1, \ldots, x_n$, the $R_{\text{abs}}^{(D,S)}$-encoding produces a CSP Θ_F^A over A with n variables $v_1, \ldots, v_n$, unary constraints $v_i \in S$, and binary constraints $(v_u, v_v) \in R_{\text{abs}}^{(D,S)}$ for each clause $(x_u \vee x_v)$ of F. This is an arity-at-most-two CSP of size linear in $|F| + n$.

Fix any $d_0 \in D$ and $s_0 \in S \setminus D$; define the lift $\sigma : \{0,1\}^n \to S^n$ by $\sigma(x)_i = d_0$ if $x_i = 1$, and $\sigma(x)_i = s_0$ if $x_i = 0$.

Lemma 12 (Generalized KC-reducibility for value-test DNNFs)**.** *Let (D, S) be a strict cube-term blocker of A, F a monotone OR-CNF on n variables, and $\mathcal{R}$ a decomposable AND/OR circuit representing $\text{Sol}(\Theta_F^A) \subseteq S^n \subseteq A^n$, in which each leaf is a* literal *of the form $v_i = a$ or $v_i \neq a$ for some $i \in [n]$ and $a \in A$. Then there is a Boolean* DNNF *D' for the satisfying-assignment set of F with $|D'| \leq |\mathcal{R}|$.*

Proof. For each Boolean assignment $x \in \{0,1\}^n$ recall $\sigma(x) \in S^n$ with $\sigma(x)_i = d_0$ if $x_i = 1$, else s_0. Define a leaf substitution from A-valued literals to Boolean literals as follows:

- $v_i = d_0 \mapsto x_i, \quad v_i = s_0 \mapsto \neg x_i, \quad v_i = a \mapsto \mathsf{FALSE}$ for $a \notin \{d_0, s_0\}$;
- $v_i \neq d_0 \mapsto \neg x_i, \quad v_i \neq s_0 \mapsto x_i, \quad v_i \neq a \mapsto \mathsf{TRUE}$ for $a \notin \{d_0, s_0\}$.

For every x, each substituted leaf evaluates to the truth value the original leaf has under $\sigma(x)$, so the resulting Boolean circuit D' computes $\mathcal{R}(\sigma(x))$. The AND/OR structure and decomposability are preserved, only leaf labels change; $|D'| \leq |\mathcal{R}|$.

By construction of $R_{\mathrm{abs}}^{(D,S)}$ under the identification $d_0 \leftrightarrow 1$, $s_0 \leftrightarrow 0$, each clause $(x_u \vee x_v)$ holds on x iff $(\sigma(x)_u, \sigma(x)_v) \in R_{\mathrm{abs}}^{(D,S)}$, so $\sigma(x) \in \mathrm{Sol}(\Theta_F^A) \iff x \models F$. Hence D' computes $\mathrm{Sat}(F)$. □

Lemma 12 strictly subsumes Lemma 5: when $S = B$ is two-element and $D = \{b_0\}$, $R_{\mathrm{abs}}^{(D,S)}$ coincides with R_{NAND} on B, and the substitution reproduces the constant-factor relabeling used in the proof of Lemma 5. The new lemma drops hypothesis (D2) entirely: leaf substitution avoids the per-coordinate domain-restriction step, so only decomposability (D1) is invoked in the proof.[2]

Encoding finite-domain solution sets as Boolean DNNF**.** The standard KC treatment of finite-domain CSPs encodes each A-valued variable v_i either by a one-hot block of $|A|$ Boolean variables $\{y_{i,a}\}_{a \in A}$ with the implicit $\mathrm{ExactlyOne}_a y_{i,a}$ constraint, or by a fixed bit-encoding using $\lceil \log_2 |A| \rceil$ Boolean variables per v_i. The substitution of Lemma 12 extends directly to Boolean DNNF representations over either encoding; we record the substitution explicitly so that the lemma can be invoked on standard DNNF inputs without first translating them.

One-hot encoding. Each leaf $y_{i,a}$ is the Boolean indicator of $v_i = a$, and $\neg y_{i,a}$ is the indicator of $v_i \neq a$. Substitute $y_{i,d_0} \mapsto x_i$, $y_{i,s_0} \mapsto \neg x_i$, and $y_{i,a} \mapsto \mathsf{FALSE}$ for $a \notin \{d_0, s_0\}$; the corresponding negated leaves $\neg y_{i,a}$ map dually to $\neg x_i$, x_i, TRUE respectively. The ExactlyOne constraint, restricted to assignments with $\sigma(x)$ as their A-valued image, is automatically satisfied.

Bit encoding. Fix the codewords $b(d_0), b(s_0) \in \{0,1\}^{\lceil \log_2 |A| \rceil}$ for d_0 and s_0. Each Boolean bit leaf of the encoding becomes either x_i, $\neg x_i$, TRUE, or FALSE according to the two bit values that $b(d_0)$ and $b(s_0)$ place in that position (agree on the bit → constant; disagree → x_i or its negation depending on which codeword places 1 in that position). On assignments inconsistent with the lift $\sigma(x)$ — those whose bit pattern at the coordinate of v_i is the codeword of a value outside $\{d_0, s_0\}$ — the relevant bit leaves are forced to constants (FALSE or TRUE depending on the bit value of that codeword in their position), and the analogous dual substitution handles negations.

In both cases the substitution preserves AND/OR structure, decomposability, and size up to an $O(|A|)$ constant absorbed into the implicit Ω in the lower bound below. Consequently the conclusion of Lemma 12 "Boolean DNNF D' for $\mathrm{Sat}(F)$ of size $\leq |\mathcal{R}|$" applies verbatim when $\mathcal{R}$ is a Boolean DNNF over either standard encoding.

7.3 Universal residual-blocker theorem

Theorem 13 (Full residual-blocker lower bound)**.** *Let A be a finite idempotent Maroti pair without a cube term. There is a family $\{\Theta_n^A\}_{n \geq 1}$ of arity-at-most-two CSP instances over A of size polynomial in n such that every* DNNF *representation of the solution set has size $2^{\Omega(n)}$. Consequently the same bound holds for OBDD, FBDD, SDD, d-DNNF, and every standard knowledge-compilation language $\mathcal{L} \subseteq$ DNNF.*

[2] Hypothesis (D3) was already inert in the proof of Lemma 5, recording a standard property of the target languages rather than driving any step of the construction; it likewise plays no role here.

Proof. By Marković–Maróti–McKenzie [16] Theorem 2.1, A has a strict cube-term blocker (D, S) with $\emptyset \neq D \subsetneq S \leq A$. By Lemma 11, $R_{\text{abs}}^{(D,S)}$ is $\text{Clo}(A)$-invariant. Take F_n to be the Bova–Capelli–Mengel–Slivovsky family of monotone 2-CNFs on bounded-degree expander graphs over n variables [3], and let $\Theta_n^A := \Theta_{F_n}^A$ be its $R_{\text{abs}}^{(D,S)}$-encoded CSP, of size linear in $|F_n| + n$ (each clause maps to one binary constraint, each variable to one unary $v_i \in S$ constraint).

Let $\mathcal{R}$ be any DNNF representing $\text{Sol}(\Theta_n^A)$ — either a decomposable AND/OR circuit with $v_i = a / v_i \neq a$ literal leaves, or a Boolean DNNF over the one-hot or bit encoding of A-valued variables. In the value-test case Lemma 12 applies directly; in the Boolean-encoding case the substitution paragraph above produces a Boolean DNNF D' for $\text{Sat}(F_n)$ with $|D'| \leq c_A|\mathcal{R}|$ for some constant $c_A = O(|A|)$ depending only on A. The BCMS lower bound [3] forces $|D'| \geq 2^{\Omega(n)}$, hence $|\mathcal{R}| \geq 2^{\Omega(n)}$ (the c_A factor is absorbed into the implicit constant in Ω). □

Corollary 14 (No universal KC compiler, residual regime)**.** *For every standard knowledge-compilation language $\mathcal{L} \subseteq$ DNNF, there is no universal polynomial-size $\mathcal{L}$-compilation scheme for the class of CSP instances of finite idempotent Maroti pairs without a cube term.*

Proof. A universal compiler would in particular handle each such A on the family $\{\Theta_n^A\}$ of Theorem 13, contradicting the $2^{\Omega(n)}$ lower bound. □

8 A Parameterized Positive Slice

The witness-family lower bound rules out universal compilation into fixed KC languages, but it does not rule out parameterized exact compilation for tame instances. The classical parameter here is treewidth of the primal graph.

Theorem 15 (Bounded-treewidth static compilation)**.** *Fix a finite algebra A. Let Θ be a binary CSP instance over A whose primal graph has treewidth at most w, and suppose a width-w nice tree decomposition is given. Then one can build, in time $|A|^{O(w)}\text{poly}(|\Theta|)$, an exact compiled object $\mathcal{T}(\Theta)$ of total size $|A|^{O(w)}\text{poly}(|\Theta|)$ consisting of:*

(a) the nice tree decomposition $(T, (B_t)_{t \in V(T)})$ itself, with constraint scopes annotated at the introduce node where they first become visible;

(b) for each bag B_t, the bag table $F_t : A^{B_t} \to \mathbb{N}$ defined in the proof sketch below, together with backpointers from each entry to the child entries that produced it.

From $\mathcal{T}(\Theta)$ one obtains:

(i) exact membership queries in polynomial time (walk the tree, check each bag table against the projection of the queried assignment);

(ii) exact counting and exact unary marginals in $|A|^{O(w)}\text{poly}(|\Theta|)$ time;

(iii) exact projection to any specified subset of variables by repeated forgetting of variables;

(iv) enumeration with polynomial delay via the standard backpointer-driven self-reduction.

The construction is the standard bounded-treewidth CSP dynamic program of Freuder [8], and more generally Courcelle's MSO-to-tree-automaton paradigm [6].

Proof sketch. Take a nice tree decomposition $(T, (B_t)_{t \in V(T)})$ of width w. For each bag B_t build a table

$$F_t : A^{B_t} \to \mathbb{N},$$

where $F_t(\alpha)$ counts the number of extensions of the partial assignment α to the variables in the subtree below t that satisfy all constraints introduced there. At an introduce node one checks local consistency of the new variable with the constraints whose scope is now fully visible. At a forget node one sums over the forgotten variable. At a join node one multiplies the two child tables with the standard correction for double counting of the shared bag assignment. Since each bag has size at most $w + 1$, every table has at most $|A|^{w+1}$ entries, and there are $O(|\Theta|)$ bags.

The stored bag tables, together with the tree decomposition, the per-bag constraint scopes, and the backpointers from each entry to the child entries that produced it, give an exact compiled object for the solution set by standard junction-tree semantics. Exact counting is obtained by summing the root table. Exact unary marginals are obtained by pinning a value and rerunning the same recurrence, or equivalently by the standard inside-outside rerooting variant. Exact projection is obtained by repeated forget operations on the tables along the decomposition. Membership is checked by walking the decomposition and verifying agreement against each bag table. Polynomial-delay enumeration follows from the same tables and backpointers by standard self-reduction. □

The hard witness family lies outside this parameterized slice for a structural reason.

Proposition 16 (BCMS witnesses have linear treewidth)**.** *Let G_n be the bounded-degree expander graphs underlying the Bova–Capelli–Mengel–Slivovsky witness family. Then* $\mathrm{tw}(G_n) = \Omega(n)$*, i.e. linear in the number n of vertices, equivalently variables. Consequently, the binary witness CSP instances Θ_n^A obtained from these graphs through the verified $R_{\mathrm{OR}}/R_{\mathrm{NAND}}$ encoding also have primal treewidth $\Omega(n)$.*

Proof sketch. Every graph of treewidth k has a balanced separator of size at most $k+1$. By contrast, a bounded-degree expander has no balanced separator of sublinear size: any separator that splits off two linear-sized pieces must itself have size $\Omega(n)$; see, for example, Grohe–Marx [10]. Therefore the expander graphs G_n have treewidth $\Omega(n)$. The CSP encoding used in the witness-family lower bound places one variable at each vertex and one binary constraint on each graph edge, so the primal graph of Θ_n^A is exactly G_n. □

Corollary 17 (Parameterized dichotomy for the witness construction)**.** *For the arity-at-most-two witness instances arising from the verified $R_{\mathrm{OR}}/R_{\mathrm{NAND}}$ encoding, bounded treewidth yields an exact $|A|^{O(w)}\mathrm{poly}(n)$ compilation by the theorem above, while the BCMS witness family has treewidth $\Omega(n)$ and is excluded from any universal polynomial-size compilation into fixed KC languages by the witness-family lower bound.*

9 Beyond Standard KC

The main theorem rules out compilation into the standard KC languages, not into every conceivable representation. The remaining possibilities fall into five targets: an exact static set representation, a broader circuit representation, a dynamic exact procedure, an approximate-access scheme, or a parameterized exact compilation. The table below records what happens to each of them on the present witness family.

Computational model	Representative candidates	Theoretical barrier on the witness family
Exact-static set representations	Maroti-specific recursive stubs, memoized residual-state DAGs, certificate/grammar schemes, polynomial and Gröbner encodings, cluster decompositions, Sims-style orbit compression	Either collapse back to the CSP itself, are blocked by BCMS witness geometry, or enter the still-open context-sensitive symbolic regime discussed after the table.
Broader circuit representations	Late-binding templates, open-term circuits, nondecomposable residual-state DAGs	Exposed-interface exact-summary tree and DAG circuits, and strict-parametric template DAGs, are all ruled out unconditionally in Section 9.1 below; the remaining open cases are context-dependent symbolic templates or growing-arity parameterizations beyond the strict-parametric regime.
Dynamic exact procedures	Bulatov–Zhuk and memoized-trace variants [5, 20]	Compatible with tractability, but not static compilation.
Approximate-access schemes	Glauber/JVV-type sampling and counting, low-degree SoS/Lasserre relaxations [13, 19, 15, 17, 9, 18]	Viable only after relaxing exactness or the query model.
Parameterized exact schemes	Bounded-treewidth dynamic programming; residual-width-style FPT compilations [8, 6]	Viable only after relaxing universality to a structural parameter.

For the BCMS-derived witness family, the first row keeps failing for the same small set of informal obstructions. *The following observations are conjectural* — they are suggested by the witness family and auxiliary computations on small n, but are not used anywhere in the formal proofs of Theorems 8 and 13. Concretely, on $\{\Theta_n^A\}$ at growing n:

(S1) the Booleanized solution sets appear to fill the full $\mathbb{F}_2$-affine span, so any encoding whose compression mechanism is linear-algebraic over $\mathbb{F}_2$ would collapse back to the original presentation;

(S2) no small clone-orbit quotient is visible, so orbit- or symmetry-based compressions do not appear to produce sublinear classes;

(S3) the vanishing ideal is naturally generated by the edge constraints plus Boolean field equations, so polynomial and Gröbner-basis encodings seem only to re-express the CSP without compressing it;

(S4) the natural Hamming-local solution graphs appear to remain (near-)single-cluster, so cluster decompositions do not look likely to split hard instances into a small catalog of separately describable components;

(S5) the host graphs are expanders, so planarity-bounded exact encodings do not apply.

Any new exact-static candidate must therefore do one of two things on the witness family: break one of (S1)–(S5), or leave the setting of Lemma 5 (equivalently Lemma 12). The next subsection sharpens this within broader-circuit models.

9.1 Unconditional lower bounds for broader circuit classes

This subsection extends the negative answer to two broader exact-static circuit classes: exposed-interface exact-summary circuits (in tree and DAG form) and strict-parametric symbolic template DAGs. For both, the witness family forces $2^{\Omega(n)}$ size unconditionally, via a fully-induced-matching counting argument on the BCMS expander cuts, transferred to the four-algebra witnesses and onward to the full residual blocker regime by restriction closure analogous to Lemma 12. The defining feature of the models is that at each gate, the summary on each side records enough information to determine the global extension behavior of partial solutions to the rest of the variables, not just across the sibling. Without that condition, a path-style circuit adding one vertex at a time would represent $\mathrm{IS}(G_n)$ with constant local summary per gate. The strengthened condition is the natural exact-static analog of the bag tables produced by the tree-decomposition compiler of Section 8.

Two broader-circuit classes.

Definition 18 (Exposed-interface summary tree circuit)**.** An *exposed-interface summary tree circuit* for a relation $R \subseteq A^n$ on variable set V is a rooted binary tree of join gates such that

(1) each leaf corresponds to a single variable $v \in V$ and represents $R|_{\{v\}}$;

(2) each internal gate g has two children on disjoint variable sets $X_g, Y_g \subseteq V$, with $V_g := X_g \sqcup Y_g$, and represents $R|_{V_g}$;

(3) *(exposed-interface refinement)* each child exposes a summary map $\lambda_{g,X} : R|_{X_g} \to \Sigma_{g,X}$, and likewise $\lambda_{g,Y}$, that refines global extension-equivalence in R: $\lambda_{g,X}(\alpha) = \lambda_{g,X}(\alpha')$ implies that for every $K \in R|_{V \setminus X_g}$, $\alpha \cup K \in R$ iff $\alpha' \cup K \in R$;

(4) *(local join compatibility)* there is a compatibility relation $K_g \subseteq \Sigma_{g,X} \times \Sigma_{g,Y}$ such that for all $\alpha \in R|_{X_g}, \beta \in R|_{Y_g}$, $\alpha \cup \beta \in R|_{V_g}$ iff $(\lambda_{g,X}(\alpha), \lambda_{g,Y}(\beta)) \in K_g$.

The size of the circuit is $\sum_g \max\{|\Sigma_{g,X}|, |\Sigma_{g,Y}|\}$.

Definition 19 (Exposed-interface summary DAG circuit)**.** An *exposed-interface summary DAG circuit* for R is obtained from Definition 18 by allowing shared subgraphs. Explicitly, every join occurrence in the unrolled decomposable structure must satisfy the exposed-interface refinement clause (Definition 18(3)) and the local join compatibility clause (4) on its own variable split. Reusable nodes are identified by literal pairs (W, σ) where $W \subseteq V$ is the exposed interface specification and $\sigma \in \Sigma$ is a global extension-equivalence class of $R|_W$; two occurrences share a node iff they expose the same W and the same realized state σ. The size of the DAG is the number of distinct (W, σ) state nodes it realizes.

Clause (3) is the model's defining content: it requires each exposed summary to encode the partial solution's full extension behavior to the remaining $V \setminus X_g$ variables, not merely its behavior across the sibling Y_g. This matches the standard bag-table notion from tree-decomposition compilation, and is also the condition under which the witness family forces exponential summary alphabets at some balanced gate.

We treat a circuit as a piece of structural data — a tree (or DAG) shape together with, at each gate, the alphabets $\Sigma_{g,X}, \Sigma_{g,Y}$, the summary maps $\lambda_{g,X}, \lambda_{g,Y}$, and the compatibility relation K_g. Clauses (3) and (4) constrain admissible data; the size measure counts alphabet cardinalities. The lower bounds below say that no admissible instance of this structure representing the witness family can have all summary alphabets simultaneously small.

Key combinatorial input. Fix a graph G on vertex set V and write $\mathrm{IS}(G)$ for the independent-set relation. For a subset $W \subseteq V$ and an independent set $I \subseteq W$, write $B_{V\setminus W}(I) := N(I) \cap (V \setminus W)$ for the *blocked set* in the complement. Two independent sets $I, I' \subseteq W$ are *globally extension-equivalent* if for every $K \in \mathrm{IS}(G[V \setminus W])$, $I \cup K \in \mathrm{IS}(G)$ iff $I' \cup K \in \mathrm{IS}(G)$.

Proposition 20 (Global interface states are blocked sets)**.** *For $I, I' \in \mathrm{IS}(G[W])$, global extension-equivalence holds iff $B_{V\setminus W}(I) = B_{V\setminus W}(I')$.*

Proof. If the blocked sets agree, then for any $K \in \mathrm{IS}(G[V \setminus W])$, $I \cup K$ is independent in G iff K avoids $B_{V\setminus W}(I)$, which is the same condition for I'. Conversely, if $y \in B_{V\setminus W}(I) \triangle B_{V\setminus W}(I')$, the singleton $K = \{y\}$ distinguishes I from I'. □

We need a matching condition strong enough that the witness sets I_S built from it are independent in G, not merely in the bipartite cut graph. A matching $\{(u_i, v_i)\}_{i=1}^k$ between W and $V \setminus W$ in G is *fully induced* if the induced subgraph $G[\{u_1, \ldots, u_k\} \cup \{v_1, \ldots, v_k\}]$ consists of exactly the matching edges $\{u_i v_i\}$: no edges among the u_i, no edges among the v_i, and no cross-edges $u_i v_j$ for $i \neq j$.

Proposition 21 (Fully induced matching forces exponential entropy)**.** *Let $\{(u_i, v_i)\}_{i=1}^k$ be a fully induced matching between W and $V \setminus W$ in G. Then the number of global extension-equivalence classes on the W side is at least 2^k.*

Proof. For each $S \subseteq [k]$ set $I_S := \{u_i : i \in S\}$. Since there are no edges among the u_i (by full induction), each I_S is independent in G. For $j \in [k]$: if $j \in S$, the matching edge $u_j v_j$ forces $v_j \in B_{V\setminus W}(I_S)$; if $j \notin S$, no $u_i \in I_S$ is adjacent to v_j (the only candidate via the matching would be u_j, but $j \notin S$; full induction rules out the other cross-edges $u_i v_j$ for $i \neq j$). So $B_{V\setminus W}(I_S) \cap \{v_1, \ldots, v_k\} = \{v_i : i \in S\}$, and distinct S yield distinct blocked sets. Apply Proposition 20. □

Proposition 22 (Linear edges give linear fully induced matching)**.** *Let G have maximum degree Δ, and let $W \subseteq V$. If $|E(W, V \setminus W)| \geq m$, there is a fully induced matching between W and $V \setminus W$ in G of size at least $m/(2\Delta - 1)^2$.*

Proof. Let $H = G[W, V \setminus W]$ be the bipartite subgraph of crossing edges. A maximal matching M in H satisfies $|M| \geq m/(2\Delta - 1)$: every edge of H shares an endpoint with some matching edge, and one matching edge uv touches at most $2\Delta - 1$ edges of H.

Form the conflict graph K on vertex set M, with two matching edges uv and $u'v'$ adjacent in K iff *any* edge of G joins an endpoint of one to an endpoint of the other — whether a same-side edge (uu' or vv') or a cross-edge (uv' or $u'v$). Each endpoint of uv has at most $\Delta - 1$ neighbors in G besides its matched partner, and each such neighbor can be the endpoint of at most one edge of M. So uv conflicts with at most $2(\Delta - 1)$ other matching edges. The maximum degree of K is therefore at most $2\Delta - 2$, and K has an independent set of size at least $|M|/(2\Delta - 1) \geq m/(2\Delta - 1)^2$. An independent set in K is precisely a fully induced matching between W and $V \setminus W$ in G. □

Lower bound for the broader-circuit classes. The BCMS witnesses are independent-set instances on bounded-degree expanders G_n of maximum degree Δ (typically $\Delta = 3$). For some constants $\alpha \in (0, 1/2]$ and $h > 0$ depending only on the family, every $W \subseteq V(G_n)$ with $\alpha n \leq |W| \leq (1 - \alpha)n$ satisfies $|E(W, V(G_n) \setminus W)| \geq hn$; this is the standard edge-expansion bound for bounded-degree expanders (see, e.g., [10]).

Theorem 23 (Exposed-interface summary tree lower bound)**.** *Every exposed-interface summary tree circuit representing $\mathrm{IS}(G_n)$ has size at least $2^{hn/(2\Delta-1)^2}$, where h is the expander balanced-cut constant of G_n for imbalance ratio up to 5:1.*

Proof. Standard tree bookkeeping gives an internal gate g with $|V_g| \in [n/3, 2n/3]$. The larger child has size at least $|V_g|/2 \geq n/6$: rename it $W := X_g$, so $n/6 \leq |W| \leq 2n/3$. Then $|V(G_n) \setminus W| \geq n/3$, and the imbalance between W and its complement is at most 5:1. By edge expansion, $|E(W, V(G_n) \setminus W)| \geq hn$. By Proposition 22, there is a fully induced matching between W and $V(G_n) \setminus W$ in G_n of size at least $hn/(2\Delta-1)^2$. By Proposition 21, $\mathrm{IS}(G_n[W])$ has at least $2^{hn/(2\Delta-1)^2}$ global extension-equivalence classes.

By Definition 18(3), $\lambda_{g,X}$ refines global extension-equivalence in $\mathrm{IS}(G_n)$ on the W side, so $|\Sigma_{g,X}| \geq 2^{hn/(2\Delta-1)^2}$. Gate g alone contributes this much to the circuit size. □

Theorem 24 (Exposed-interface summary DAG lower bound)**.** *Every exposed-interface summary DAG circuit representing* $\mathrm{IS}(G_n)$ *has size at least* $2^{hn/(2\Delta-1)^2}$.

Proof. The same bookkeeping locates a join occurrence in the underlying decomposable structure with $|V_g| \in [n/3, 2n/3]$ and identifies a side W of size at least $n/6$. The same expansion argument gives $2^{hn/(2\Delta-1)^2}$ global extension-equivalence classes on W. By Definition 19, for the fixed exposed interface W, distinct exact summary states $\sigma \neq \sigma'$ yield distinct DAG nodes (W, σ), (W, σ'); and each realized equivalence class requires its own summary state, since two classes mapping to the same σ would violate the refinement clause. So the DAG has at least $2^{hn/(2\Delta-1)^2}$ distinct nodes. □

Restriction closure and transfer.

Lemma 25 (Restriction closure)**.** *Let $B \subseteq A$ and $R \subseteq A^n$. Given an exposed-interface summary tree circuit (or DAG circuit) for R, the following construction yields a circuit of the same class for $R \cap B^n$ of weakly smaller size:*

- *replace each leaf domain A by B;*
- *at each gate g, restrict $\lambda_{g,X}$ to the smaller domain $R|_{X_g} \cap B^{X_g}$, with new summary alphabet $\Sigma^B_{g,X} := \lambda_{g,X}(R|_{X_g} \cap B^{X_g}) \subseteq \Sigma_{g,X}$;*
- *restrict the compatibility relation K_g to $\Sigma^B_{g,X} \times \Sigma^B_{g,Y}$.*

Proof. The new summary alphabet $\Sigma^B_{g,X}$ is the image of $\lambda_{g,X}$ on the restricted domain, hence a subset of $\Sigma_{g,X}$, and $|\Sigma^B_{g,X}| \leq |\Sigma_{g,X}|$. The refinement clause is preserved inside $R \cap B^n$: if two partial solutions $\alpha, \alpha' \in R|_{X_g} \cap B^{X_g}$ have $\lambda_{g,X}(\alpha) = \lambda_{g,X}(\alpha')$, then by Definition 18(3) applied in R they agree on extensions to every $K \in R|_{V \setminus X_g}$, hence in particular on every $K \in (R \cap B^n)|_{V \setminus X_g}$. Local join compatibility restricts analogously. In the DAG case, distinct nodes $(W, \sigma) \neq (W, \sigma')$ may now collapse if both summaries land in the same restricted image, but no node splits, so the node count weakly decreases. □

Theorem 26 (Broader-circuit lower bounds, four-algebra witnesses)**.** *For each witness algebra $A \in \{A_{\wedge,5}, A_{\vee,5}, A_{\wedge,7}, A_{\vee,7}\}$, every exposed-interface summary tree circuit and every exposed-interface summary DAG circuit for* $\mathrm{Sol}(\Theta^A_n)$ *has size* $2^{\Omega(n)}$.

Proof. The verified two-element subuniverse $B \subseteq A$ identifies $\mathrm{Sol}(\Theta^A_n) \cap B^n$ with $\mathrm{IS}(G_n)$ (via R_{NAND}, or its bit-complement via R_{OR}). Apply Lemma 25 to restrict to B^n, then Theorem 23 or 24. □

Theorem 27 (Broader-circuit lower bounds, residual blocker regime)**.** *Let A be a finite idempotent Maroti pair without a cube term. There is a family $\{\Theta^A_n\}_{n \geq 1}$ of arity-at-most-two CSP instances over A of size polynomial in n such that every exposed-interface summary tree circuit and every exposed-interface summary DAG circuit for* $\mathrm{Sol}(\Theta^A_n)$ *has size* $2^{\Omega(n)}$.

Proof. Use the $R_{\mathrm{abs}}^{(D,S)}$-encoded family $\Theta_n^A := \Theta_{F_n}^A$ of Theorem 13 with F_n the BCMS expander 2-CNF on G_n. Apply Lemma 25 with $B = \{d_0, s_0\} \subseteq S$; under $d_0 \leftrightarrow 1, s_0 \leftrightarrow 0$, this identifies $\mathrm{Sol}(\Theta_n^A) \cap B^n$ with $\mathrm{IS}(G_n)$ and reduces any representation to one on $\mathrm{IS}(G_n)$ of weakly smaller size. Apply Theorem 23 or 24. □

Strict-parametric template DAGs.

Definition 28 (Strict-parametric template DAG)**.** A *strict-parametric template DAG family* for a sequence of relations $\{R_n\}_{n\geq 1}$ on variable sets V_n consists of: a family-fixed symbolic alphabet $\Sigma_{\mathrm{sym}}(n)$ of size $n^{O(1)}$; a family-fixed parameter alphabet $\mathrm{Param}(n)$ of size $n^{O(1)}$; an integer $k = O(1)$; a family-fixed resolution map

$$\rho_n : \Sigma_{\mathrm{sym}}(n) \times \mathrm{Param}(n)^k \to \Sigma_{\mathrm{exact}}(n);$$

and, for each n, an exposed-interface summary DAG circuit (Definition 19) for R_n in which each reusable node stores only a symbolic state $\tau \in \Sigma_{\mathrm{sym}}(n)$, while parent context supplies a parameter tuple $(p_1, \ldots, p_k) \in \mathrm{Param}(n)^k$, the exact summary exposed at the interface in that parent context being $\rho_n(\tau, p_1, \ldots, p_k)$.

The lower bound below depends only on the realizable image of ρ_n at each fixed exposed interface, not on any specific evaluation discipline for parameters: regardless of whether $(p_1, \ldots, p_k)$ is computed by a bottom-up sweep, supplied by the parent gate, or threaded along the DAG by any other rule, the set of exact summary states reachable at a given interface lies inside $\rho_n(\Sigma_{\mathrm{sym}}(n) \times \mathrm{Param}(n)^k)$, of size $n^{O(1)}$. The constraint $k = O(1)$ is therefore the bottleneck: once k grows, the polynomial parameter alphabet starts to index exponentially many tuples and the counting argument breaks.

Lemma 29 (Polynomial summary count per interface)**.** *In a strict-parametric template DAG family, at most $|\Sigma_{\mathrm{sym}}(n)| \cdot |\mathrm{Param}(n)|^k = n^{O(1)}$ distinct exact summary states are realizable at any fixed exposed interface.*

Proof. The set of realizable exact summaries at a fixed interface is contained in $\rho_n(\Sigma_{\mathrm{sym}}(n) \times \mathrm{Param}(n)^k)$, whose cardinality is bounded by the domain size. □

Theorem 30 (Strict-parametric exclusion)**.** *For each witness algebra $A \in \{A_{\wedge,5}, A_{\vee,5}, A_{\wedge,7}, A_{\vee,7}\}$, and more generally for every finite idempotent Maroti pair A without a cube term, there is no strict-parametric template DAG family representing $\{\mathrm{Sol}(\Theta_n^A)\}$ for all sufficiently large n.*

Proof. By Theorem 26 (resp. Theorem 27), some balanced join on $\mathrm{Sol}(\Theta_n^A)$ requires at least $2^{\Omega(n)}$ distinct exact summary states. By Lemma 29, a strict-parametric template DAG realizes at most $n^{O(1)}$ exact summary states at any fixed cut. For all sufficiently large n, $n^{O(1)} < 2^{\Omega(n)}$. □

Open frontier. After Theorems 27 and 30, the remaining symbolic-circuit escape is narrow: any candidate must allow at least one of (i) parameter arity $k = \omega(1)$, (ii) context-dependent resolution maps that vary with the parent context outside a family-fixed alphabet, or (iii) super-polynomial symbolic or parameter alphabets. Each of these places the candidate strictly outside both broader-circuit classes covered above, and any one suffices to escape the counting argument; we single this regime out as the natural next direction. In screening exercises against the witness corpus, every concrete candidate we have considered either collapses into one of the classes already covered above or pays the (i)–(iii) cost without delivering actual compression on $\{\Theta_n^A\}$. The frontier is narrow in practice, even though it is not formally closed.

Dynamic, approximate, and parameterized alternatives also remain open, but they change the target rather than competing with universal exact-static compilation. In that sense the broader frontier has not disappeared, but it is much narrower and more specific than "anything outside KC."

10 Conclusion

The algebraic-CSP literature supplies powerful static representations when a fixed term identity is available. Maroti pairs are tractable, but they lie partly outside those standard static-template regimes. This paper shows that the gap is real on two scales. First, for four explicit witness algebras (Theorem 8), bounded-arity CSP solution sets already require exponential size in every standard knowledge-compilation language contained in DNNF; this alone suffices to rule out a universal polynomial-size $\mathcal{L}$-compiler for any $\mathcal{L} \subseteq$ DNNF across all finite idempotent Maroti pairs. Second, the witness construction extends uniformly to the entire residual blocker regime (Theorem 13, Corollary 14): every finite idempotent Maroti pair without a cube term admits a polynomial-size arity-at-most-two CSP family whose DNNF representation requires size $2^{\Omega(n)}$, via the $R_{\mathrm{abs}}^{(D,S)}$ encoding on a strict cube-term blocker carrier.

This does not rule out parameterized positive results. In particular, bounded-treewidth instances still admit the standard exact dynamic-programming compilation, whereas the BCMS witness family lies outside that slice because its primal graphs are expanders of linear treewidth.

Beyond standard KC, the picture sharpens rather than disappears. Two broader exact-static circuit-representation tiers — exposed-interface exact-summary circuits (in both tree and DAG form) and strict-parametric symbolic template DAGs — admit the same $2^{\Omega(n)}$ lower bound on the same witness families, via a direct fully-induced-matching counting argument across balanced expander cuts (Section 9.1). The remaining symbolic escape is narrow: any surviving candidate must use either growing parameter arity, context-dependent resolution maps, or super-polynomial symbolic alphabets. We single out genuinely context-sensitive resolution as the natural next direction. Dynamic, approximate, and parameterized alternatives stay open as well, but they change the target rather than competing with universal exact-static compilation. On the present witness corpus, this remaining escape is open in principle but empirically inert: every standard exact-static candidate screened so far falls under the lower bounds above. Genuine progress likely requires either a fundamentally different target (amortized, advice-extended, or instance-specific compilation) or a different witness corpus on which natural compressors find structure to exploit.

Acknowledgments

AI assistants Claude Opus 4.7 and GPT-5.4 contributed to literature search, brute-force verification scripts for the witness algebras, LaTeX drafting, and proof refactoring during the development of this paper. The author takes full responsibility for the technical content and conclusions.

A Witness Catalog

The appendix records the four witness algebras used in the lower-bound construction. The data actually used in the proof are the verified two-element subuniverses and the restrictions of p and q to those subuniverses. The full operation signatures are included only as machine-readable reproducibility data; they are not intended for manual inspection by the reader.

The verified two-element subuniverses are:

Witness	Two-element subuniverse	Verified invariant relation
$A_{\wedge,5}$	$\{0,4\}$	R_{NAND}
$A_{\vee,5}$	$\{2,3\}$	R_{OR}
$A_{\wedge,7}$	$\{0,1\}$	R_{NAND}
$A_{\vee,7}$	$\{4,6\}$	R_{OR}

For $A_{\wedge,5}$ the restricted generators are literally Boolean AND. For $A_{\vee,5}$ and $A_{\vee,7}$, the restricted operation $q|_B$ is Boolean OR and $p|_B$ preserves R_{OR} by direct check. For $A_{\wedge,7}$ the proof uses a dominant-projection check on the restricted generators.

A.1 Restricted witness tables used in the proof

In each row below, the tuple order on B^3 is the lexicographic order (b_0,b_0,b_0), (b_0,b_0,b_1), (b_0,b_1,b_0), (b_0,b_1,b_1), (b_1,b_0,b_0), (b_1,b_0,b_1), (b_1,b_1,b_0), (b_1,b_1,b_1).

Witness	B	$p\|_B$ outputs	$q\|_B$ outputs	Consequence
$A_{\wedge,5}$	$\{0,4\}$	`00000004`	`00000004`	Boolean AND; preserves R_{NAND}.
$A_{\vee,5}$	$\{2,3\}$	`22333333`	`23333333`	$q\|_B$ is Boolean OR; $p\|_B$ preserves R_{OR} by direct check.
$A_{\wedge,7}$	$\{0,1\}$	`00001001`	`00010011`	Direct dominant-projection check preserves R_{NAND}.
$A_{\vee,7}$	$\{4,6\}$	`44666666`	`46666666`	$q\|_B$ is Boolean OR; $p\|_B$ preserves R_{OR} by direct check.

These restricted tables are the only operation data used in the transfer to binary CSPs. In particular, the verification of the invariant binary relation for each witness is a direct coordinatewise check on these eight tuples, not a manual reconstruction from the full signatures.

A.2 Machine-readable full signatures

Each ternary operation $f \in \{p,q\}$ on domain $A = \{0,1,\ldots,|A|-1\}$ is recorded as the concatenated string of values $f(x_1,x_2,x_3)$ taken in lexicographic order over $(x_1,x_2,x_3) \in A^3$: position $|A|^2 \cdot x_1 + |A| \cdot x_2 + x_3 + 1$ in the string is the value $f(x_1,x_2,x_3)$. So for $|A| = 5$ the string has 125 digits and for $|A| = 7$ it has 343 digits, in both cases over the alphabet $\{0,\ldots,|A|-1\}$. The Maroti identities (M1)–(M3) and the verified-subuniverse invariance of R_{NAND} or R_{OR} can be checked independently by reading the strings into an $|A| \times |A| \times |A|$ table under this convention and running a few-line script.

Witness $A_{\wedge,5}$. Domain size 5; verified two-element subuniverse $\{0,4\}$; verified invariant relation R_{NAND}.

$p =$ `031000030333130344430142042132012343342440311410013033143000102202402400431224104033401144443302031304120441202114300111024 14`

$q =$ `001400421333331330240334002102414111334040102210441234324012332213001334421433120113303241201331021103310311404213322331334 1332233104114`

Witness $A_{\vee,5}$. Domain size 5; verified two-element subuniverse $\{2,3\}$; verified invariant relation R_{OR}.

$p =$ 013332004220321342202240113313414032202113231440141214411214442204333212243130322330332331343311233213113103222132114242134244

$q =$ 003212112420133340122024103040110344214233434143203313000204112332133401213232340330340330133323003410003444024130111021103440

Witness $A_{\wedge,7}$**.** Domain size 7; verified two-element subuniverse $\{0, 1\}$; verified invariant relation R_{NAND}.

$p =$ 005402200124314236426140163114204204031334253025010112210166200056331321112304452116263664432022653361155352342330211364346615340652244561121154002505466610505264162464106306611332304545136533126243246640611041025433120052201233411616054654521506135005240510101430630610504215415453500535625101422646636415616003225405164264602035220462212513600

$q =$ 003143001464634630035161561015044004214066224064400310621161145555050131001643406653664544642212663460100262403235265124546020050352164565414456210166424451203266162313506323203035313325102130113141326635103022635532515222314655440151051610331323125502642620046102666643301354345164401514345006011146635632413625005310214226630332516312460126600

Witness $A_{\vee,7}$**.** Domain size 7; verified two-element subuniverse $\{4, 6\}$; verified invariant relation R_{OR}.

$p =$ 054631616502523623152311311602612322024560244634344230341161351444365030043115630252601126351452350200450131531306236233254434601442245606320361654256063036621465565216253334615153545665306111520335564006454235120600365261234464041321614354165642033353442424303513024165344313415522555631111405203615564046135652554426201460061626114152451362763643333262206053102323342241310564214164144265643633510660514614444301634551113462106361541403503244344456645411401446661365364656334060334122146045634266330310052453265465333561625614523213203444106511336253402416162464621666266451636160152604325344400534001011044364500351041342254525203210443265462342126056462216305566661600

References

[1] K. A. Baker and A. F. Pixley. Polynomial interpolation and the Chinese remainder theorem for algebraic systems. *Mathematische Zeitschrift*, 143:165–174, 1975.

[2] Libor Barto and Marcin Kozik. Constraint satisfaction problems solvable by local consistency methods. *Journal of the ACM*, 61(1):3:1–3:19, 2014.

[3] Simone Bova, Florent Capelli, Stefan Mengel, and Friedrich Slivovsky. Expander CNFs have exponential DNNF size. *SIAM Journal on Computing*, 46(5):1605–1630, 2017.

[4] Andrei A. Bulatov and Victor Dalmau. A simple algorithm for Mal'tsev constraints. *SIAM Journal on Computing*, 36(1):16–27, 2006.

[5] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 319–330, 2017.

[6] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12–75, 1990.

[7] Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.

[8] Eugene C. Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proceedings of the 8th National Conference on Artificial Intelligence*, pages 4–9, 1990.

[9] Dima Grigoriev. Linear lower bound on degrees of positivstellensatz calculus proofs for the parity. *Theoretical Computer Science*, 259(1–2):613–622, 2001.

[10] Martin Grohe and Dániel Marx. On tree width, bramble size, and expansion. *Journal of Combinatorial Theory, Series B*, 99(1):218–228, 2009.

[11] David Hobby and Ralph McKenzie. *The Structure of Finite Algebras*. American Mathematical Society, 1988.

[12] Paweł Idziak, Petar Marković, Ralph McKenzie, Matthew Valeriote, and Ross Willard. Tractability and learnability arising from algebras with few subpowers. *SIAM Journal on Computing*, 39(7):3023–3037, 2010.

[13] Mark Jerrum, Leslie G. Valiant, and Vijay V. Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[14] Donald E. Knuth. The Art of Computer Programming, Volume 4B, Fascicle 7: Constraint Satisfaction. Addison-Wesley, 2025.

[15] Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[16] Petar Marković, Miklós Maróti, and Ralph McKenzie. Finitely related clones and algebras with cube terms. *Order*, 29(2):345–359, 2012.

[17] Pablo A. Parrilo. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis, California Institute of Technology, 2000.

[18] Grant Schoenebeck. Linear level Lasserre lower bounds for certain k-CSPs. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, pages 593–602, 2008.

[19] Dror Weitz. Counting independent sets up to the tree threshold. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 140–149, 2006.

[20] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. In *Proceedings of the 58th IEEE Symposium on Foundations of Computer Science*, pages 331–342, 2017.